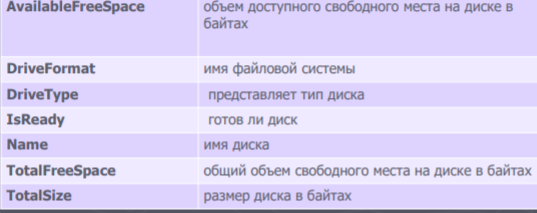
1. **Какие классы содержаться с System.IO?**

****

1. **Для чего используются классы Directory и DirectoryInfo? В чем отличие?**

Класс **Directory**. Объявлен и определен в корне System.IO. Содержит в себе группу статических членов для создания, удаления, перемещения и переименования каталогов и подкаталогов. Запечатанный и статический тип: не может иметь наследников, объект класса создать нельзя.

Класс **DirectoryInfo**. Ссылочный тип, наследуется от System.IO.FileSystemInfo, поэтому имеет массу перегруженных методов родительского класса, а также десяток полезных свойств при работе с каталогами

1. **Для чего используются классы File и FileInfo? Какие методы они содержат.**

С их помощью мы можем создавать, удалять, перемещать файлы, получать их свойства и многое другое.

Некоторые полезные методы класса **FileInfo:**

* **CopyTo(path)**: копирует файл в новое место по указанному пути path
* **Create()**: создает файл
* **Delete()**: удаляет файл
* **MoveTo(destFileName)**: перемещает файл в новое место

Класс **File** реализует похожую функциональность с помощью статических методов:

* **Copy()**: копирует файл в новое место
* **Create()**: создает файл
* **Delete()**: удаляет файл
* **Move**: перемещает файл в новое место
* **Exists(file)**: определяет, существует ли файл

1. **Для чего используются классы StreamReader и StreamWriter?**

**9 вопрос**

1. **Для чего используются классы BinaryWriter и BinaryReader?**

**9 вопрос**

1. **Как можно сжимать и восстанавливать файлы?**

С помощью классов **ZipFile, DeflateStream** и **GZipStream**, которые находятся в пространстве имен **System.IO.Compression**

**ZipFile:** предоставляет статические методы для создания, извлечения и открытия zip-архивов.

**DeflateStream:** предоставляет методы и свойства для сжатия и распаковки потоков с помощью алгоритма *Deflate.*

**GZipStream:** предоставляет методы и свойства, используемые для сжатия и распаковки потоков с использованием спецификации формата данных **GZip**.

1. **Расскажите алгоритм сжатия GZip.**

1.создать поток для чтения из исходного файла.

2.перенести поток для записи в сжатый файл.

3.инициализировать набор архивации.

4.получить данные из потока

1. **Для чего служит класс Path?**

Предназначен для работы с именами файлов и путями в файловой системе

*Назначение*

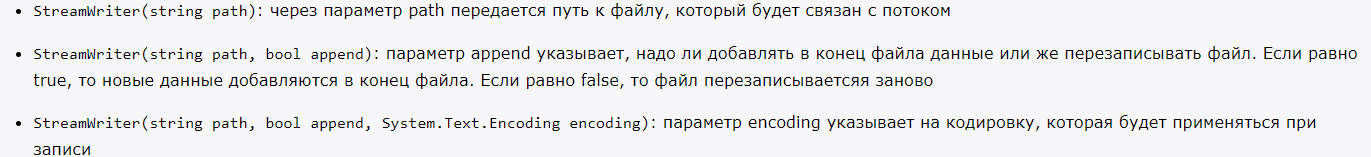
* выделить имя файла из полного пути
* скомбинировать для получения пути имя файла и имя каталога
* сгенерировать имя для временного файла или каталога

1. **Что такое тестовый и бинарный файл. Как можно выполнить чтение и запись в эти файлы**

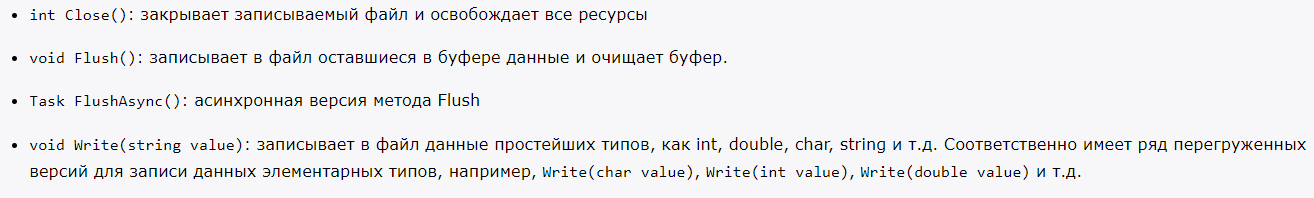
**Текстовый файл.**

**StreamWriter** – для записи в текстовый файл

Конструкторы:

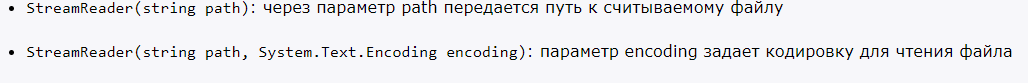


Методы:

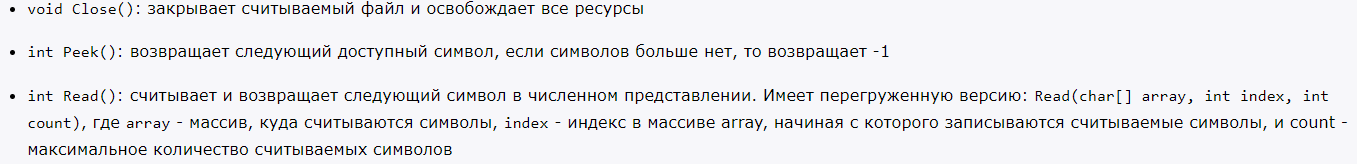


**StreamReader** - считывать весь текст или отдельные строки из текстового файла.

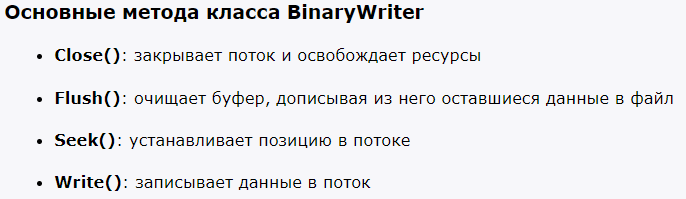
Конструкторы:

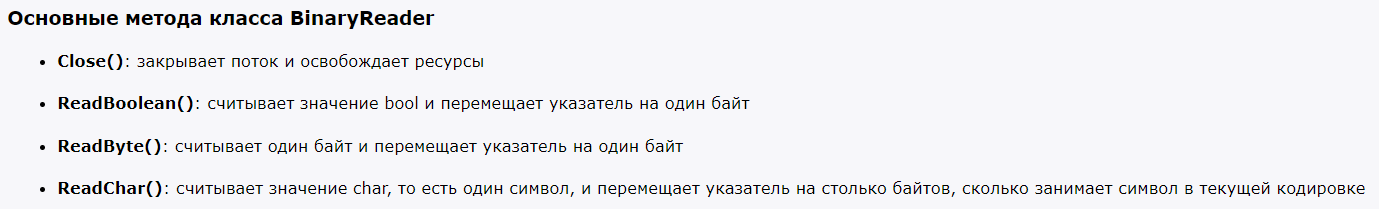


Методы:



**Бинарный файл** (последовательность произвольных байтов.)**.**

****

****

1. **Что такое произвольный доступ к файлу? Приведите пример.**

Произвольный доступ к файлу - это многократное чтение и многократное изменение своих данных в процессе одного сеанса работы

1. **Как применяется конструкция using (не директива) при работе с файловыми потоками? Для чего ее используют.**

**using** - конструкция для автоматического вызова метода Dispose (в котором при реализации интерфейса в классе должно происходить освобождение неуправляемых ресурсов).

Конструкция using оформляет блок кода и создает объект некоторого класса, который реализует интерфейс IDisposable, в частности, его метод Dispose. При завершении блока кода у объекта вызывается метод Dispose.

